

CoMOGrad and PHOG: From Computer Vision to Fast and Accurate Protein Tertiary Structure Retrieval

Rezaul Karim¹, Mohd. Momin Al Aziz¹, Swakkhar Shatabda², M. Sohel Rahman¹,
Md. Abul Kashem Mia¹ and Farhana Zaman¹, Salman Rakin¹

¹Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology,
Palashi, Dhaka-1000, Bangladesh.

²Department of Computer Science and Engineering,
United International University,
House 80, Road 8A, Dhanmondi, Dhaka-1209, Bangladesh.

Abstract

Due to the advancements in technology number of entries in the structural database of proteins are increasing day by day. Methods for retrieving protein tertiary structures from this large database is the key to comparative analysis of structures which plays an important role to understand proteins and their function. In this paper, we present fast and accurate methods for the retrieval of proteins from a large database with tertiary structures similar to a query protein. Our proposed methods borrow ideas from the field of computer vision. The speed and accuracy of our methods comes from the two newly introduced features, the co-occurrence matrix of the oriented gradient and pyramid histogram of oriented gradient and from the use of Euclidean distance as the distance measure. Experimental results clearly indicate the superiority of our approach in both running time and accuracy. Our method is readily available for use from this website: <http://research.buet.ac.bd:8080/Comograd/>.

1 Introduction

Proteins perform a large array of functions within a living organism. Proteins are polymers of amino acid monomers. Linear sequence of amino acids is called the primary structure of a protein. There are three other levels of structural complexity: secondary, tertiary and quaternary. Secondary structure is the local spatial arrangement of the backbone atoms formed by intramolecular and intermolecular hydrogen bonding of amide groups. Tertiary structure refers to the three-dimensional structure of an entire polypeptide and quaternary structure is the spatial arrangement of two or more polypeptide

chains known as sub-units.

Every protein has a unique, stable and kinetically accessible [1] three dimensional structure or tertiary structure also known as the native structure. The functionalities of a protein are principally correlated to its tertiary structure. That is why a protein ceases to function when its tertiary structure is broken by denaturation at a high temperature; although, its primary structure may still remain intact [21]. Proteins with similar tertiary structure have similar ligand binding sites and pockets [22]. Moreover, tertiary structures are more conserved than amino acid sequence during evolution [11]. Misfolded proteins are the cause of many critical diseases like Alzheimer’s disease [6]. Therefore, analysis and study of similar tertiary structures is of great importance in function prediction of novel proteins, study of evolution, disease diagnosis, drug discovery, antibody design and many other fields.

Following the determination of the first tertiary structure in 1958 using crystallography [12], biologists have successfully determined a large number of structures by now. Due to recent developments in X-ray crystallography and nuclear magnetic resonance imaging, there has been a rapid increase in the number of experimentally determined structures stored in the world-wide repository, Protein Data Bank (PDB) [3] (<http://wwpdb.org/>). RCSB PDB is the primary repository for known protein structures. As of February 11, 2014, the number of protein structures stored in PDB was more than 97,789. With this increase in the number of known protein structures, the need for fast and accurate algorithms for the retrieval of protein tertiary structures is now greater than ever. In the protein tertiary structure retrieval problem, the task is to retrieve proteins having most similar structures from the known structure database. The results are ranked based on similarity or distance measures.

There exist numerous approaches in the literature that focus on fast and accurate retrieval of protein tertiary structures. The traditional way to compare structures is to treat each one as a rigid three-dimensional object and superimpose one on the other. Differences are then calculated using different distance metrics, e.g., least-squares method. In the pattern recognition literature, the structures are often represented by feature vectors and similarity or dissimilarity is measured by comparing the feature vectors with one another. As a feature to represent the tertiary structure of a protein chain, the α carbon distance matrix is widely used by many researchers [20, 10]. Here, the 3D coordinate data is mapped to a two dimensional feature matrix. The α carbon distance matrix gives the intra-molecular distances of α carbons in a protein chain. This α carbon distance matrix resembles the tertiary structure of a protein and the secondary structure elements conserved in it. Notably, α carbon distance matrix based exact algorithms run in $O(N!)$ time assuming that the input matrix is of size

$N \times N$.

DALI [10] compares structures by aligning the α carbon distance matrices. Each distance matrix is decomposed into sub-matrices of fixed size which is called the elementary contact patterns. It compares those contact patterns (pair-wise), and store the matching pairs in a list with a matching score. Then, it assembles pairs in the correct order using Monte Carlo optimization to yield the overall alignment and the final matching score. In a later version, it has used branch and bound method to assemble the pairs [9]. The *CE* method [19] also takes the α carbon distance matrix as the feature vector and uses combinatorial extension and Monte Carlo optimization to compare protein structures in a way much similar to *DALI*. Both *DALI* and *CE* require lots of computation and hence the corresponding web servers respond to the query requests via email only after a certain period that is required for costly processing. A faster approach based on the α carbon distance matrix as a feature is *MatAlign* [2]. It provides an $O(N^4)$ dynamic programming solution where N is the dimension of the distance matrix. *MatAlign* compares distance matrix of the query protein and the target protein row by row and builds up a dynamic programming table based on a row by row matching score. Then, it aligns rows of the query protein with the rows of the target protein to maximize the corresponding matching score. The higher is the matching score, the more is the similarity in the corresponding structures.

Marsolo et al. [14] have introduced a wavelet based approach that resizes the distance matrices of the protein structures before the actual comparison is done. Later on Mirceva et al. have introduced *MASASW* [15] that uses wavelet coefficients of distance matrices as the feature vector. It has been shown in [15] that *Daubechies-2 wavelets* improves accuracy than others. *MASASW* transforms all the α carbon distance matrices into a 32×32 matrix by interpolation and wavelet transformation. It then compares them like *MatAlign* but with a sliding window to reduce the number of comparisons. The time complexity of *MASASW* is $O(wWN^2)$ where w and W are window sizes and N is the dimension of the distance matrices. As has been mentioned above, *MASASW* assumes $N = 32$.

Despite that several methods are found in the literature for protein structure retrieval, the quest for even faster and more accurate methods still continues as the number of known protein structures is growing very fast. In this paper, we present an extremely fast and highly accurate novel method of retrieving proteins with similar tertiary structures from a large database. In particular, here we present an ultra fast algorithm based on two novel feature vectors. These are the **Co**-occurrence **M**atrix of the **O**riented **G**radient of **D**istance Matrices (CoMOGrad) and **P**yramid **H**istogram of **O**riented **G**radient (PHOG). Additionally, as will be reported later, our proposed algorithm gives more accurate results

than the state of the art methods. Very briefly, much of the speed and accuracy we have achieved comes from the introduction of the novel features from the field of computer vision and pattern recognition. Our aim has been to introduce a feature which does not require any complex algorithm to compare the tertiary structures. Rather, a simple distance measure to calculate the distance between the two vector quantities is used in our approach. As has already been mentioned above, some previous methods in the literature have used the α carbon distance matrix as a feature vector. Upon analyzing the tertiary structure and α carbon distance matrix represented as a gray-scale image, we have observed that not all data in the matrix seem to be equally important. We further have realized that the co-occurrence matrix (as will be reported later) of the oriented gradient of the distance matrix is the most important feature with respect to the comparison of tertiary structures. Finally, we have found that the Euclidean distance or ℓ^2 norm of our novel features as the distance measure outperforms the widely used costly alignment distance/similarity measure of α carbon distance matrices. As a result, the combination of the above ideas gives us an extremely fast method without sacrificing the accuracy.

Later in this paper, we first describe our proposed approach along with the novel features. Following this, we report the experimental results and relevant discussion. Finally, we conclude the paper with an outline of the future works.

We have carefully analyzed the gray-scale images from the α carbon distance matrices and the tertiary structures. We have observed that the α helices and the anti-parallel beta sheets appear as dark lines parallel to the diagonal dark line and parallel beta sheets appear as dark lines normal to the diagonal dark line. Beta sheets of two strips appear as one dark line normal to the diagonal; beta sheets of three strips appear as two dark lines normal to the diagonal and one dark line parallel to the diagonal. In general, for a standard beta sheet, the number of points of co-occurrence of parallel and anti-parallel diagonal lines depends on the number of strips in the beta sheets. Again, the number of single parallel lines depends on the number of standard α helices. Moreover, length of those lines near the diagonal region is proportional to the length of the α helices. The distance of the parallel lines from the diagonal dark line is proportional to the radius of the α helix. Figure 1 depicts the corresponding α carbon distance matrix as a gray scale image of a tertiary structure of a protein with beta sheets. In the gray scale image, the 7 anti-parallel dark lines near the diagonal dark line correspond to the presence of 8 beta sheets in the corresponding protein structure and the lengths of those dark lines are proportional to the lengths of the beta sheets. Figure 2 represents the gray scale image of the corresponding α carbon distance matrix of a protein tertiary structure with two alpha helices. Here in

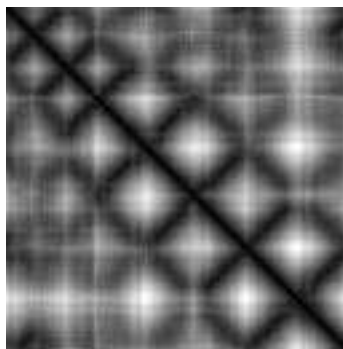


Figure 1: Representation of β sheets of domain dln4ja [13] in α carbon distance matrix gray-scale image.

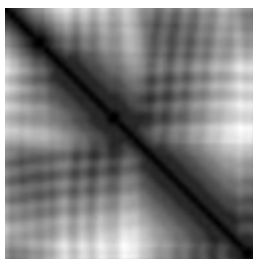


Figure 2: Representation of α helices of domain dlirqa [16] in α carbon distance matrix gray-scale image.

the image, the two parallel dark lines near the diagonal dark line correspond to the presence of two α helices in the protein structure and the lengths of the dark lines are proportional to the lengths of the α helices.

In the contemporary literature of computer vision and digital image processing [8] the lines in digital images are usually recognized from the gradient of the images. This leads us to believe that, the co-occurrence of gradient angles represents the secondary structure elements more precisely than just the distance matrix image. The tertiary structure involves the presence of secondary structure elements (SSEs), their size and position in the chain and their orientation. From the images of protein structures and their corresponding α carbon distance matrix gray-scale image, it is clear that the SSEs are represented by the orientation of dark lines at the near diagonal region of the α carbon distance matrix image. The position of the SSEs in a protein chain is represented by the position of the dark lines at the near diagonal region of the image. Here, *near diagonal region* is the region nearby the diagonal dark line. The size of the SSEs are represented by the length of the dark line at the near diagonal region. The orientation of the SSEs are represented by the presence of the dark lines at the far diagonal regions and the darkness and orientation of those dark lines. Here, *far diagonal*

region is the region in the image that is distant from the main diagonal dark line. Therefore, we need to incorporate the gradient orientation angle along with the gradient magnitude and gradient spatial orientation to incorporate the orientation of the SSEs in the feature vector. We also need co-occurrence of gradient angles. Based on our study we introduce **Co**-occurrence **M**atrix of the **O**riented **G**radient (CoMOGrad) as our first feature vector which incorporates gradient orientation angles and co-occurrence of the gradient orientation angles. This feature enables implementation of an algorithm that can facilitate more than 100 times faster comparison than *MASASW*. However, this speed is achieved at the cost of a slight decrease in the accuracy. Subsequently, we introduce another feature vector, namely, the **P**yramid **H**istogram of **O**riented **G**radient (PHOG) by incorporating the spatial information and gradient magnitude. The combination of both CoMOGrad and PHOG features in comparing structures results in an algorithm that is not only more accurate but also more than 40 times faster than *MASASW* in comparing two structures. The details of the experiments are presented later in the results section.

2 Methods and Materials

2.1 Feature vectors and feature extraction.

Mapping 3D coordinates to 2D function: Distance matrix of α carbons in residues is a good candidate to transform the 3D structure to the corresponding 2D vector representation as shown in *MASASW* [15] and the wavelet based approach by Marsolo et al. [14]. This distance matrix gives the pairwise distance between all pairs of α carbons in the polypeptide chain. Proteins with similar tertiary structures will have similar distance matrices and vice versa. As stated earlier, if we consider the matrix as a monochromatic image, α -helices and parallel β -sheets will appear as dark lines parallel to the main diagonal and antiparallel beta sheets will appear as dark lines normal to the main diagonal. This distance matrix feature also has a very appealing property, i.e., this is translation, scaling and rotation invariant.

Interestingly, as it is a two dimensional matrix like a digital image we can easily apply image processing and computer vision algorithms on it. Similarly, we can use this matrix as an adjacency matrix and interpret it as a graph. Subsequently, graph theory techniques may also be applied to solve the tertiary structure retrieval problem with this feature. In this paper, we apply ideas from the field of image processing and computer vision. Most recently these ideas have got their niche in pedestrians

and car detection [23, 18].

Scaling $C\alpha$ - $C\alpha$ distance matrix images to the same dimension

Bi-cubic interpolation: As different protein chains have different number of α carbons, the dimensions of their α carbon distance matrices vary. Therefore, we need to scale the distance matrices to the same dimension. For scaling the distance matrices, we use the methods of digital image processing used for image resizing. At first, we scale all the images to the dimension that is a power of 2 and nearest to their original dimension. As an example, if the image dimension is 80×80 we scale it to 64×64 and if the original image dimension is 100×100 we scale to 128×128 . For this step, we use bi-cubic interpolation.

Wavelet transform: After scaling the images as mentioned above, we apply wavelet transform to transform all the images to the same dimension. Notably, wavelet transform is the most widely used technique for image scaling or resizing in digital image processing. Using wavelet transform, we scale all images to 128×128 dimension as most of the images in the previous step were found to be of that dimension. For wavelet filter we used Daubechies-2 wavelet [5] as this filter has been shown to have outperformed other traditional wavelets for protein structure feature representation by MASASW [15]. Wavelet transform of an image gives four images, namely, the *approximate detail*, *horizontal detail*, *diagonal detail* and *vertical detail*. Each of these images has dimension that is half of the original image. We take the approximate detail to scale a large image to a smaller size since this is the approximate sub sampled image. For the images with dimension greater than 128×128 , we perform wavelet transform on each images multiple times to get the approximate coefficient of size 128×128 . For images with dimension less than 128×128 , we first perform wavelet transform on the images. Then using bi-cubic interpolation, we scale all four coefficients to twice their initial size. After that, with inverse wavelet transform on the four scaled wavelet coefficients, we get original image scaled up to twice its initial size. With repeated application of scaling up (down) the images that are smaller (higher) than dimension 128×128 , we finally get all of them in the desired dimension of 128×128 .

Novel features from scaled $C\alpha$ - $C\alpha$ distance matrix images

Co-occurrence matrix of oriented gradient (CoMOGrad): After having all the images of dimension 128×128 we extract our CoMOGrad feature. First we take the gradient of each of the images and compute the gradient angle and magnitude. As the angle values are continuous quantities,

we have quantized those values. For quantization, we tuned the number of quantization bins as a parameter. With experiments using various bin sizes (9, 16, 32 etc), we have found that using 16 bins with bin size 22.5 degree gives excellent results. After quantization to 16 bins, we compute co-occurrence matrix which is a 16×16 matrix. We convert this 16×16 matrix to a vector of size 256. This is our CoMOGrad feature vector. With this feature, we can simply take Euclidean distance to compare structures rather than using the alignment technique of α carbon distance matrices used by *MASASW* and *MatAlign*. Clearly, introduction of this feature makes the comparison method much simpler and faster.

Pyramid histogram of oriented gradient (PHOG): The use of CoMOGrad gives us an ultra fast structure retrieval algorithm. However it achieves this speed at the cost of some reduction in accuracy. From the discussion in the previous sections, it is clear that we have to incorporate the gradient magnitude and spatial orientation of gradient along with the angular orientation of gradient to accurately describe the tertiary structure of a protein. The CoMOGrad feature only includes angular orientation of gradient and co-occurrence of angular orientation of gradient. To incorporate the gradient magnitude and spatial orientation of gradient along with angular orientation of gradient, we take another feature named pyramid histogram of oriented gradient (PHOG) together with our CoMOGrad feature to improve the accuracy. PHOG was first proposed by Bosch et al. [4] and successfully used in object classification and pattern recognition. We create a quad tree of the original image with the original image at the root as follows. Each node of the quad tree has four children, namely, *top-left*, *top-right*, *bottom-left* and *bottom right*. Each of these images are of size one fourth of the original image. In Figure 3, we have shown a quad tree up to level 1. In our experiments, we have taken the quad tree up to level 3 and achieved excellent results. For quad tree up to level 3, there are $1+4+4*4+4*4*4=85$ nodes. We create gradient orientation histogram with 9 bins each with 40 degree range for each of the nodes. Now, we have $85*9=768$ features. We incorporate these 768 features to a vector of size 768. Then, we normalize the vector by dividing it with the sum of its 768 components. This is our PHOG feature vector. Now, PHOG combined with CoMOGrad gives a total of $256+768=1021$ features.

Distance Measure

We use the Euclidean distance or ℓ^2 norm as the distance measure of our new features. PHOG combined with CoMOGrad can be seen as a vector of length 1021. Suppose, f_q and f_i denote the feature vectors of the query protein q and a protein i in the database, respectively. Then the distance

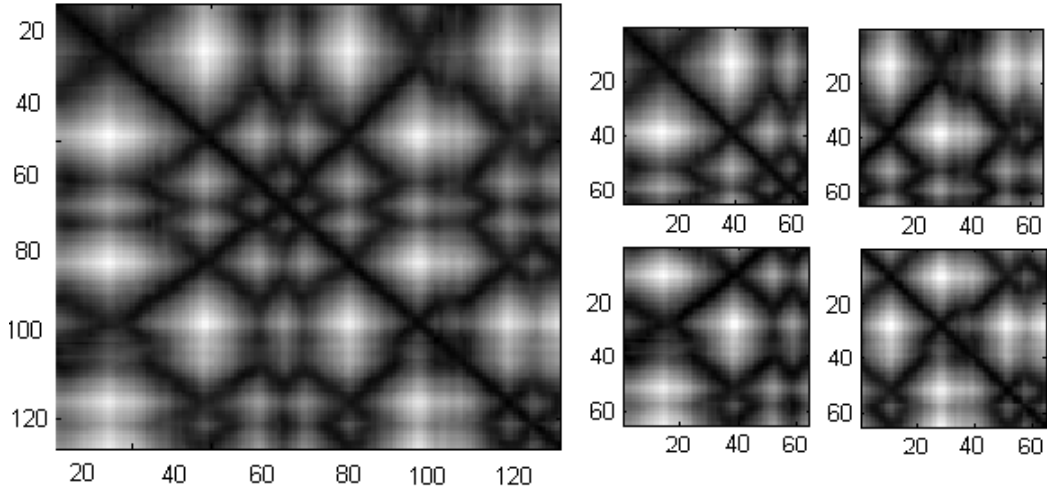


Figure 3: Level 1 quad tree of α carbon distance matrix image.

score d_i^q of protein q and i would be calculated according to Eqn. 1 below.

$$d_i^q = \sqrt{\sum_{j=1}^{1021} (|f_q[j] - f_i[j]|)^2} \quad (1)$$

Clearly, the above distance measure can be calculated in $O(N)$ time where $N = 1021$ is the size of our feature vector. Also, note that for CoMOGrad alone, $N = 256$ only. Our algorithm needs to compute d_i^q for each protein i in the database and then sort the results to rank them. As will be reported later, the ℓ^2 norm distance of the CoMOGrad and PHOG gives us a fast and much accurate retrieval algorithm.

3 Results and Discussion

Our method is readily available for use from this website: <http://research.buet.ac.bd:8080/Comograd/>. We have implemented our algorithm in Java (jdk 1.6) with Netbeans IDE and MySQL database. The feature extraction has been done Using *MATLAB* R2012. Our source code of our implementation is available from https://github.com/rezaulnkarim/protein_tertiary_structure_retrieval-. The experiments have been run on GNU/Linux debian ubuntu i686 operating system. We have used a machine having *Intel* (R) Core(TM) i5 3470 CPU 3.20 GHz with 4GB RAM. We have compared our methods with the tertiary structure retrieval method, *MASASW* [15], which is shown

to be the best performer in the literature to date. We have implemented two versions of our method, one using CoMOGrad and the other combining CoMOGrad with PHOG.

3.1 Benchmark Datasets.

For the experiments, we have used *SCOP* [7, 17] domains classification. *SCOP* is well accepted as the benchmark in the literature. We have taken 152,487 chains from *SCOP* domain as the search space and extracted features from them. As the query proteins, we have used 4965 protein chains from 417 *SCOP* family, 330 superfamily, 234 folds and 11 classes. We have run all 4965 queries over the search space of 152,487 proteins using our methods and sorted them all in ascending order based on our distance measure. For MASASW [15], we ran the same experiment and sorted results in descending order based on its similarity measure. Then we compare the results considering the top four *SCOP* labels, namely *class*, *fold*, *superfamily* and *family*. List of the query sequences are given in the Supplementary Table 1.

3.2 Accuracy Comparison.

The comparison of the accuracy of the query results among *MASASW* and our two methods are provided in Figure 4. In the line graphs plotted in this figure, the horizontal axis entitled “*number of top results*” tracks the number of top ranked query results. The vertical axes in each figure entitled “*% of ‘label’ match*” indicates the average number of query results that have matched the *SCOP* ‘label’ with the corresponding query protein. So, each point in the figure reports the average number query results that have matched the *SCOP* ‘label’ with the corresponding query protein for a specific number of top results. We report the results for all the *SCOP* labels, i.e., *class*, *fold*, *superfamily* and *family*.

As an example, suppose, we consider the top 50 results of the query and have run a total of 3 queries. Assume that, among the query results, the numbers of match for the label *family* are 40, 42 and 48 respectively. Then the average number of family match is $(40+42+48)/3=45$ and the percentage of family match for the top 50 retrieval results is $(45 \times 100) / (50) = 90$ percent. The line graphs are drawn with percentage of matches for class, fold, superfamily and family for the top 5 to top 50 retrieval results based on the selected 4965 queries.

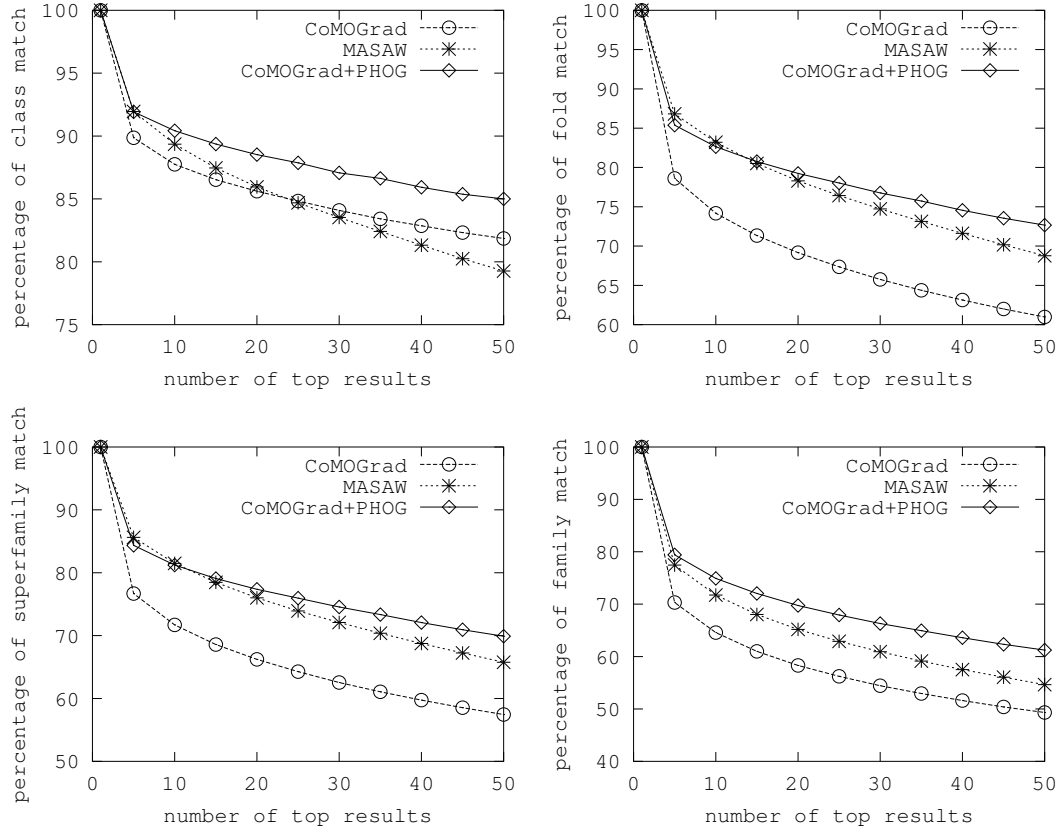


Figure 4: Percentage of matches of Class, Fold, Superfamily and Family for up to top 50 retrieval results.

Table 1: Comparison of query time.

Method	Loading Time	Query Time	Time Per Query
MASASW	28 min 11 s	42 min 18 s	25.38 s .
CoMOGrad	18 m 31s	1 m 23	0.83 s
CoMOGrad + PHOG	27 min 24 s	5min 32 s	3.32 s.

3.3 Runtime Comparison .

The time required to retrieve the protein tertiary structures for all three methods are reported in Table 1. The run time is recorded by executing 100 queries for each of the methods. In the table, *Loading Time* is the time needed to load the feature vectors from disk to memory which is done once when the system starts. The *Time Per Query* in the table is the time needed to compare the feature vector of a query protein with that of all the 152,487 proteins in our protein database and sort the results based on the distance/similarity measure and to return the sorted top ranked results. Note that the query time reported excludes the loading time. The *Query Time* in the table indicates the total time needed for 100 query structures. The results indicate that the query time for the variant with only the CoMOGrad feature is ultra fast albeit at the cost of some reduction in the accuracy. However, the variant using both CoMOGrad and PHOG as features is both super fast and more accurate than *MASASW*.

3.4 Discussion.

The exact algorithm for the matching of tertiary structures with α carbon distance matrix runs in $O(N!)$ time assuming that the input matrix is of size $N \times N$. The time complexity of *MASASW* for comparing two features is $O(wWN^2)$, which assumes $N = 32$ as the dimension of the distance matrix. Here, W and w are the size of the sliding windows to align matrices and to align rows, respectively. The authors of *MASASW* have empirically obtained the most reasonable and suitable values for W and w which are 5 and 8 respectively. Our CoMOGrad feature is a vector of size 256 and the combination of CoMOGrad and PHOG gives us a feature vector of size of 1021. For both of our methods, the run time for comparing two features is just $O(N)$. Therefore, when only the CoMOGrad feature is used, the time to compare two features is approximately $(32 \times 32 \times 5 \times 8)/(256) = 160$ times faster than *MASASW*. And, the combination of CoMOGrad and PHOG is approximately $(32 \times 32 \times 5 \times 8)/(1021) = 40$ times faster than *MASASW* in this respect. The feature extraction of the query protein as and when it is submitted as a coordinate file in the *PDB* format does not have noticeable effect on the

running time as this operation is done for one, i.e., the query protein only; the features of all the proteins in the target search space (i.e., in the database) are made available beforehand as they have already been preprocessed.

Compared to *MASASW*, the query time of the variant with CoMOGrad is almost 30 times and the variant with CoMOGrad and PHOG is more than 7 times faster. As seen in our earlier discussion, theoretically the variant with CoMOGrad is 160 times faster and the variant with CoMOGrad and PHOG is 40 times faster in comparing two features. However, in addition to the feature comparison, the actual retrieval algorithm needs to perform a sorting operation on the results of the distance/similarity values of all the proteins in the database against the query protein. Note that both *MASASW* and our two methods need to use this sorting algorithm. *MASASW* however sorts in descending order because it uses a similarity measure. As a result the actual improvement in the query time achieved by our methods does not completely match with the theoretical deduction.

In terms of accuracy, the performance of our method using only the CoMOGrad feature is almost similar to that of *MASASW*. Nonetheless, compared to *MASASW*, the combination of CoMOGrad and PHOG achieves a higher degree of accuracy. The experimental dataset contains proteins taken from 417 various families and the number is significant. The most significant achievement of the novel features we have used is the substantial reduction of the query processing time. Together the accuracy and reduced query processing time makes the structural comparison the simplest state of the art technique.

Conclusion

In this paper, we have presented two novel features, namely, CoMOGrad and PHOG, for faster and accurate retrieval of the protein tertiary structures. We have compared our results considering all the levels of *SCOP* classification hierarchy. We have reported average percentage of matching for class, fold, super family and family of our retrieval results with the query protein while most of the works in the literature have only shown similarity on only class and fold; very few in fact have worked on automated similarity match for the lower levels. Our results are in good compliance with the *SCOP* classification. CoMOGrad feature is ultra fast as compared to the state of the art methods but this extreme speed is achieved at the cost of a slight reduction in the accuracy. The combination of CoMOGrad with PHOG is also very fast and at the same time is superior in terms of accuracy as compared to state of the art methods. This creates the possibility to implement a web based service

for the protein tertiary structure retrieval with a truly online behaviour, i.e., that can provide the results in seconds, while the present web services usually provide query results via email only.

References

- [1] C. B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.
- [2] Z. Aung and K.-L. Tan. MatAlign: precise protein structure comparison by matrix alignment. *Journal of bioinformatics and computational biology*, 4(06):1197–1216, 2006.
- [3] H. M. Berman, J. D. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000.
- [4] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408. ACM, 2007.
- [5] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 41(7):909–996, 1988.
- [6] A. S. DeToma, S. Salamekh, A. Ramamoorthy, and M. H. Lim. Misfolded proteins in alzheimer’s disease and type ii diabetes. *Chemical Society Reviews*, 41(2):608–621, 2012.
- [7] N. K. Fox, S. E. Brenner, and J.-M. Chandonia. SCOPe: Structural classification of protein-sextended, integrating scop and astral data and classification of new structures. *Nucleic acids research*, 42(D1):D304–D309, 2014.
- [8] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001.
- [9] L. Holm and C. Sander. Mapping the protein universe. *Science*, 273(5275):595–602, 1996.
- [10] L. Holm and C. Sander. Dali/FSSP classification of three-dimensional protein folds. *Nucleic acids research*, 25(1):231–234, 1997.
- [11] K. Illergård, D. H. Ardell, and A. Elofsson. Structure is three to ten times more conserved than sequencea study of structural response in protein cores. *Proteins: Structure, Function, and Bioinformatics*, 77(3):499–508, 2009.

- [12] J. C. Kendrew, G. Bodo, H. M. Dintzis, R. Parrish, H. Wyckoff, and D. Phillips. A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature*, 181(4610):662–666, 1958.
- [13] I. Le Trong, S. Freitag, L. A. Klumb, V. Chu, P. S. Stayton, and R. E. Stenkamp. Structural studies of hydrogen bonds in the high-affinity streptavidin-biotin complex: mutations of amino acids interacting with the ureido oxygen of biotin. *Acta Crystallographica Section D: Biological Crystallography*, 59(9):1567–1573, 2003.
- [14] K. Marsolo, S. Parthasarathy, and K. Ramamohanarao. Structure-based querying of proteins using wavelets. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 24–33. ACM, 2006.
- [15] G. Mirceva, I. Cingovska, Z. Dimov, and D. Davcev. Efficient approaches for retrieving protein tertiary structures. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 9(4):1166–1179, July 2012.
- [16] K. Murayama, P. Orth, A. B. de la Hoz, J. C. Alonso, and W. Saenger. Crystal structure of ω transcriptional repressor encoded by streptococcus pyogenes plasmid pSM19035 at 1.5 Å resolution. *Journal of Molecular Biology*, 314(4):789 – 796, 2001.
- [17] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247(4):536 – 540, 1995.
- [18] H. Ren, C.-K. Heng, W. Zheng, L. Liang, and X. Chen. Fast object detection using boosted co-occurrence histograms of oriented gradients. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2705–2708. IEEE, 2010.
- [19] I. N. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering*, 11(9):739–747, 1998.
- [20] A. P. Singh and D. L. Brutlag. Hierarchical protein structure superposition using both secondary structure and atomic representations. In *Ismb*, volume 5, pages 284–293, 1997.
- [21] C. Tanford et al. Protein denaturation. *Adv. Protein Chem*, 23(121):282, 1968.

- [22] M. N. Wass and M. J. Sternberg. Prediction of ligand binding sites using homologous structures and conservation at CASP8. *Proteins: Structure, Function, and Bioinformatics*, 77(S9):147–151, 2009.
- [23] T. Watanabe, S. Ito, and K. Yokoi. Co-occurrence histograms of oriented gradients for pedestrian detection. In *Advances in Image and Video Technology*, pages 37–47. Springer, 2009.

Acknowledgements

The authors thanks Georgina Mirceva for sharing the implementation details of the MASASW algorithm.